DAQ software

Philip Rodrigues

University of Oxford

December 11, 2019

Outline

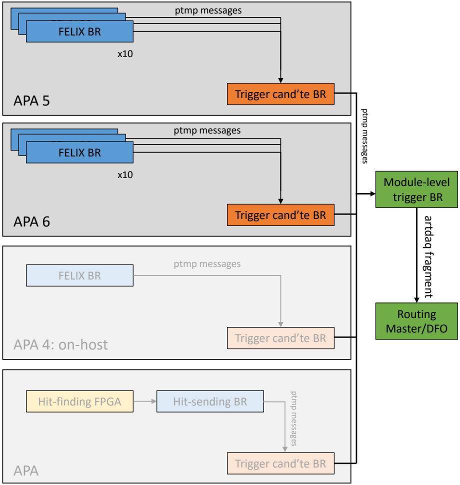
- Software-based TPC self-trigger prototype at ProtoDUNE
- Moving beyond the TDR DAQ design: online induction-wire hit finding
- Future work

ProtoDUNE self-trigger

TPC-based self-trigger: why and how

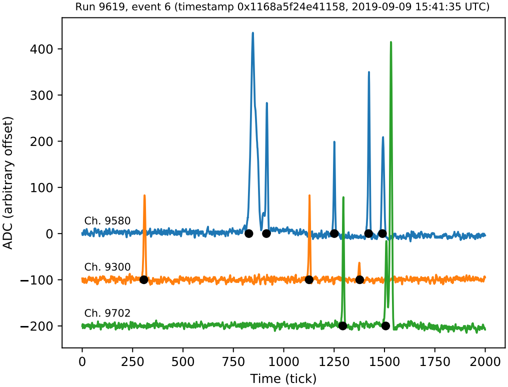
- Why: Stepping stone to DUNE FD, which will *need* a self trigger
- How:
  - Proof-of-concept; more concerned with data flow than physics
  - Make the simplest thing that works; iterate if necessary
  - Work in the existing ProtoDUNE DAQ, with incremental changes
  - Downstream (lower data rates) shouldn't impose requirements on upstream (higher data rates, performance more important)

System overview



- Implement code inside `artdaq` board readers:
  - Advantages: integration with run control, log file handling, saving of raw data and metadata
- Trigger primitives and candidates are sent using (from Brett Viren: `https://github.com/brettviren/ptmp`)
  - Data structures
  - Message passing
  - Algorithms
- Not shown: SSPs don’t participate in generating triggers, but will provide data in response to a trigger request

Trigger primitive (hit) finding

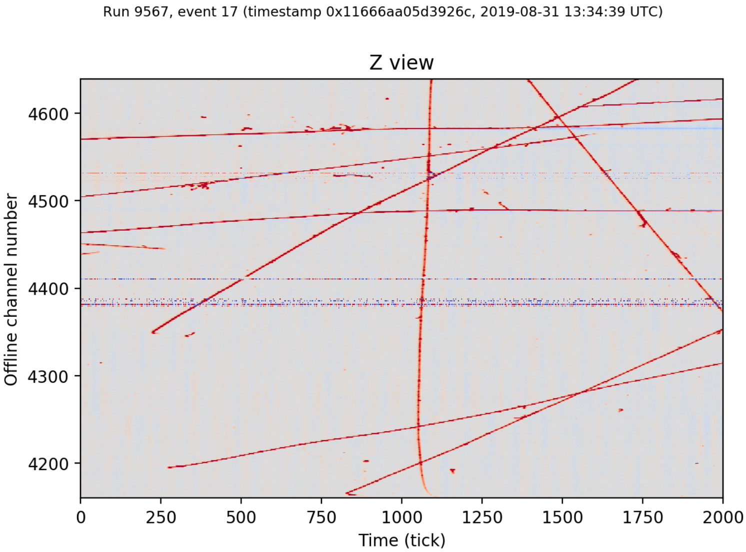


- Simple hit finding running in CPUs on FELIX BR hosts

1. Decode WIB format, select collection channels
2. Find pedestal and pedestal variance
3. Apply finite impulse response noise filter
4. Sum charge above threshold

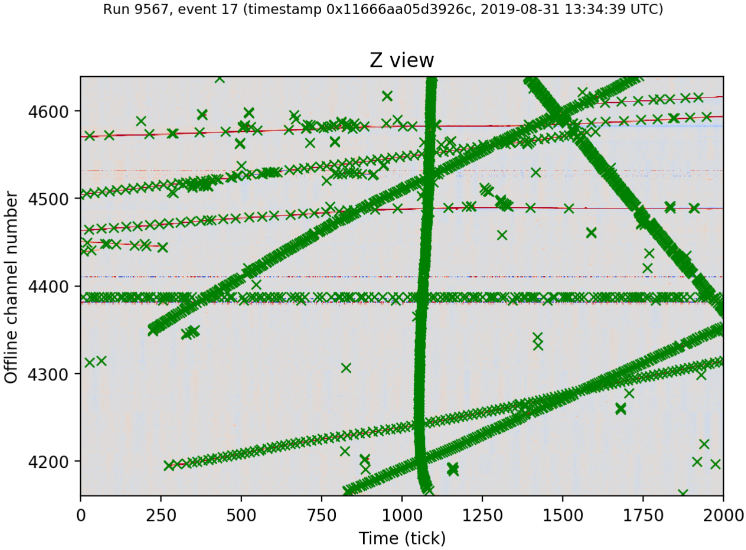
- Per APA ($\sim$ 10 GB/s): 10 CPU cores at 60% each

Hit finding works!



- Hits found continuously; read out from buffer in response to trigger

Hit finding works!



- Hits found continuously; read out from buffer in response to trigger

Trigger candidate (cluster) finding and "module"-level trigger

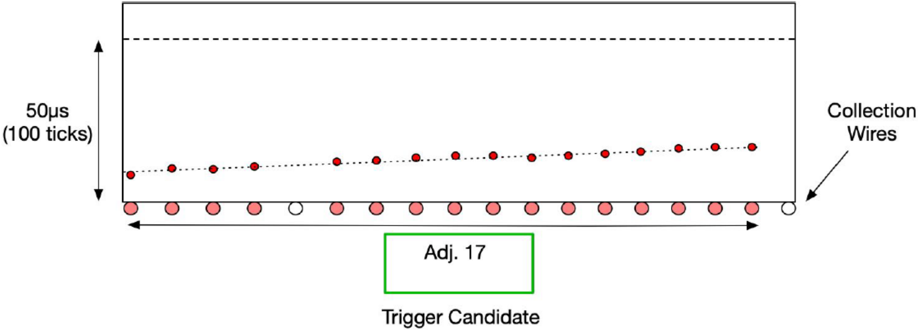
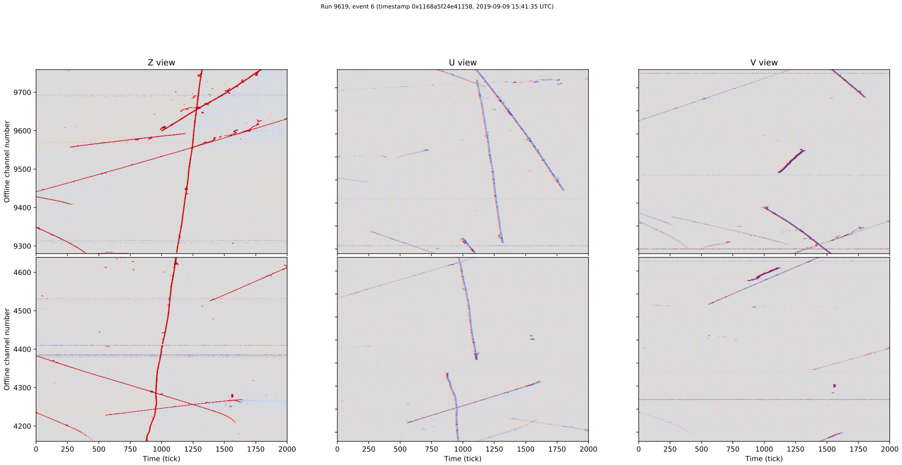


Figure: Jon Sensenig

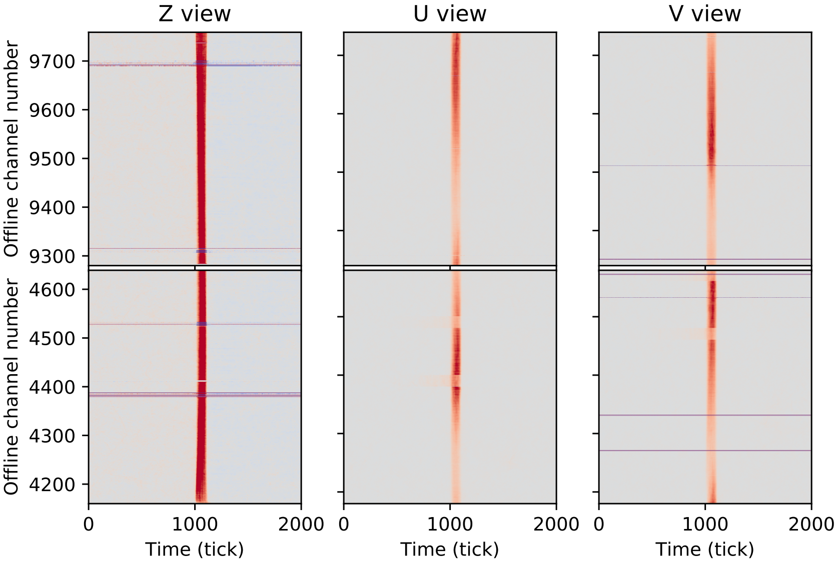
- Take 50 μs windows, find groups of hits contiguous in channel (with up to 4-wire gap)
- Generate TC if channel range of largest group is 100 or more
- Stitch together TCs from individual APAs
- Make a sliding window of one drift time, look for consistent slope (Δ(time)/Δ(channel)) between TCs
- Trigger if stitched TCs result in 350 hits or more in each APA

It works!



- Read out 1000 ticks before and after the trigger time

Sum of events



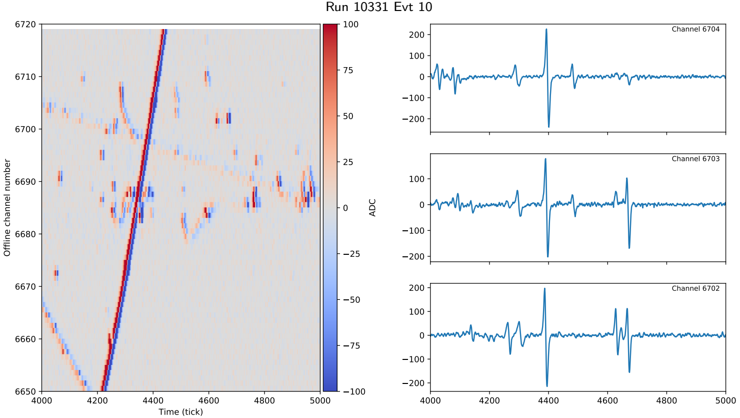
- Sum waveforms in many events. (Idea from Jon Sensenig, U. Penn)

Induction-wire hits

Background

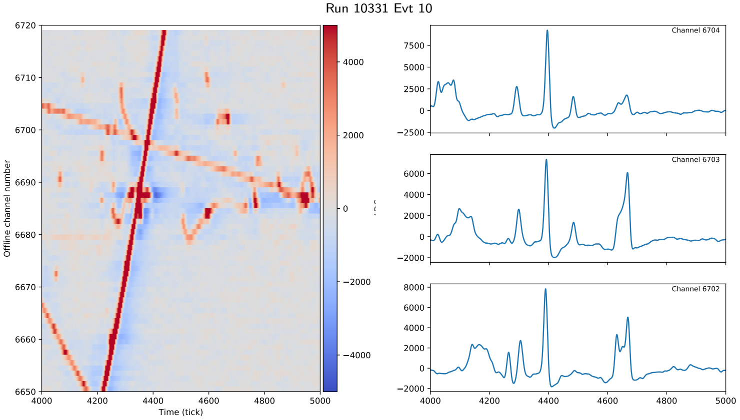
- TDR baseline design: hit-finding in collection wires only
  - Higher signal-to-noise
  - Unipolar signal easier to identify
- Possible advantages of adding induction hits-finding:
  - Full 3D reconstruction: fiducial cut
  - More information for PID (eg lowest energy solar/SN vs radioactive decays)
  - Improvements in certain noise scenarios

A sample event with “disappearing” induction signal



- Left: 2D ADCs, Right: a few channels in 1D
- The showery ~horizontal track at channels 6685–6705 is very faint: that's what we're interested in recovering
- (I've subtracted coherent noise similarly to how it's done in the offline, just to remove one extra complication. At each time tick, calculate the median value in each FEMB. Subtract that value. I suspect this can even be done efficiently *online* with a sorting network: see, eg http://pages.ripco.net/~jgamble/nw.html)

FFT deconvolution *à la* offline

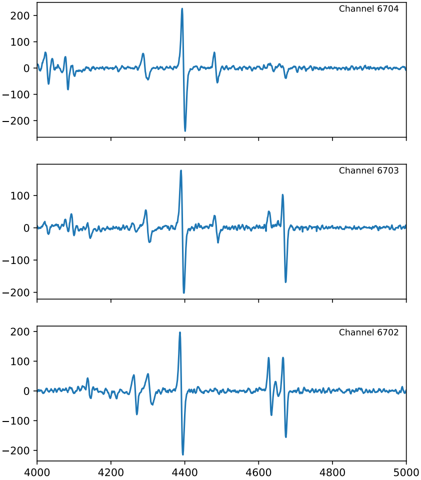


- The same after FFT-based deconvolution, similar to what's done offline in ProtoDUNE. Full details in backups
- I think this is probably not feasible to do online, but it gives an idea of what we're trying to achieve

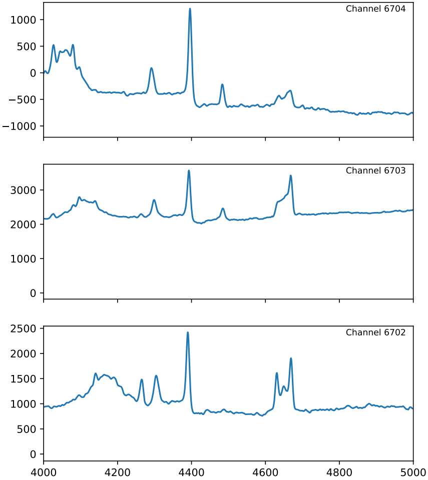
Running sum signal processing 1

- Elegantly simple idea from Francesco Pietropaolo and Filippo Resnati: just keep a running sum S_i of induction channel values. Converts bipolar to unipolar, and looks like a deconvolution

Before running sum

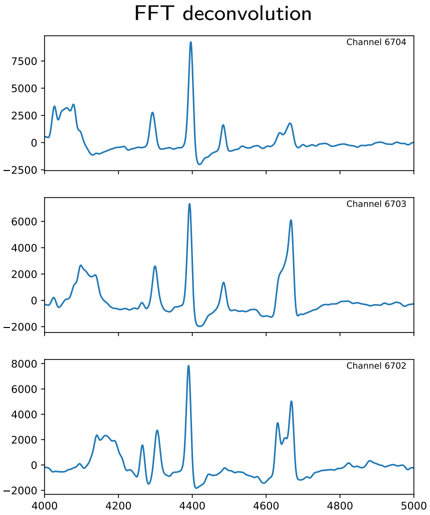


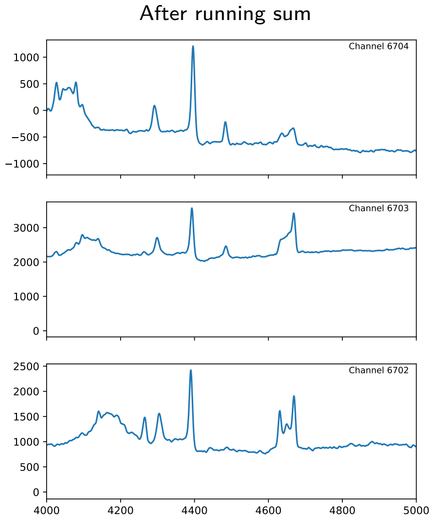
After running sum



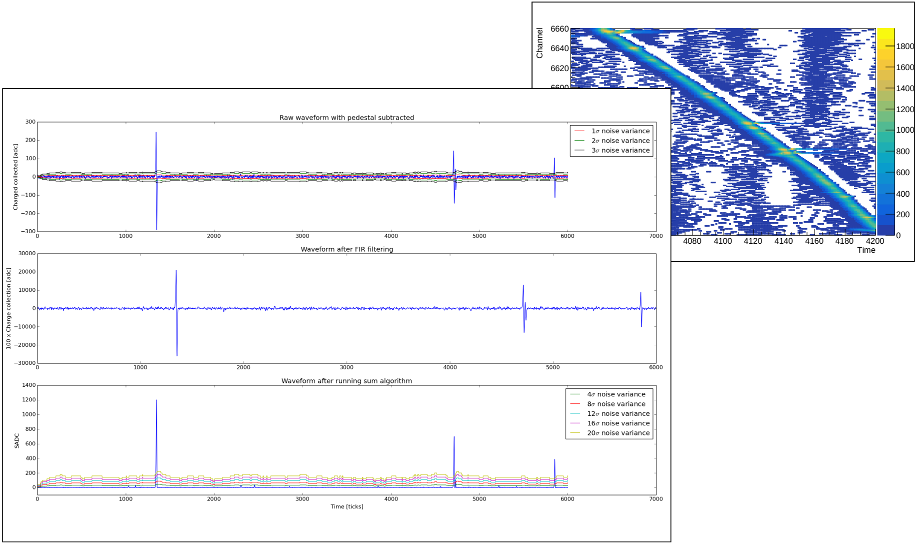
Running sum signal processing 1

- Elegantly simple idea from Francesco Pietropaolo and Filippo Resnati: just keep a running sum S_i of induction channel values. Converts bipolar to unipolar, and looks like a deconvolution





Extending hit-finding to induction wires



- Aran Borkum working on putting this into offline framework for efficiency studies

Future things

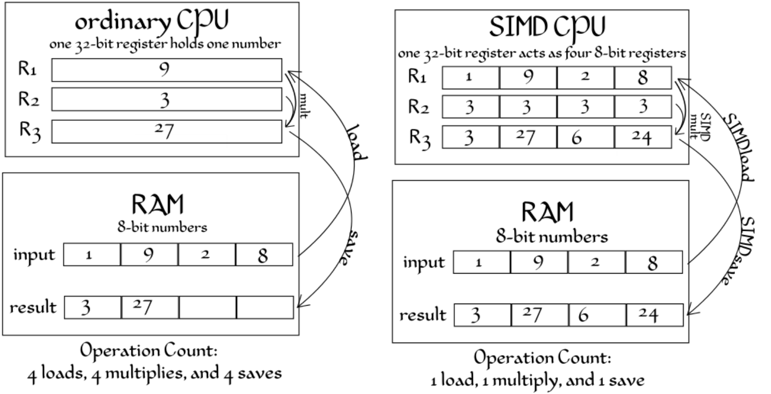
- Meeting high-uptime requirements: fault tolerance, alternative strategies for run control?
- Event building: any gain from distributed filesystems/object stores?
- Demonstrate this all at ProtoDUNE-II in 2022

Backup slides

ptmp data structures, message passing and algorithms

- `TrigPrim`: TPC hit with channel, start time, time span, ADC sum, peak ADC (unused so far), error flags (unused so far)
- `TPSet`: a container for `TrigPrims`, with count, detector ID, creation time, time/channel span, actual list of `TrigPrims`
- `TPSets` can be passed as ZeroMQ messages over network or in-process. Fast, configurable, alternatives for handling backpressure (drop or wait)
- `TPWindow` algorithm repackages `TrigPrims` into fixed-time windows
- `TPZipper` algorithm aggregates multiple `TPSet` message streams in time order, with (soft) maximum latency guarantee

SIMD

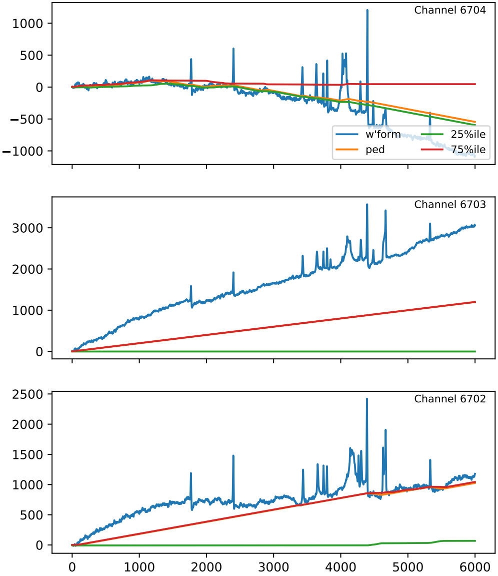


Credit: Decora at English Wikipedia. CC Attribution-Share Alike 3.0

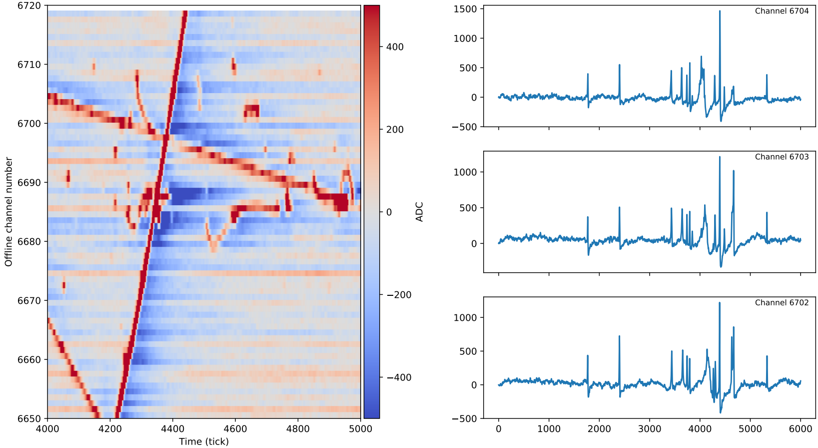
- Act on multiple values simultaneously in one instruction
- Machines I can access have AVX2 with 256-bit registers, ie 16 16-bit numbers at a time
- Now our back-of-envelope looks better: $16 N_{\text{core}}$ clock cycles per sample
- Just got access to a system at CERN with AVX-512

Running sum signal processing 2

- Running sum output, zoomed out. Large changes in pedestal
- Orange line is my “frugal streaming” estimate of the pedestal: can’t keep up
- There are probably ways to deal with this, but I haven’t thought of them yet



Running sum signal processing 3



- Francesco and Filippo point out that you can deweight the previous sum value to reduce the pedestal variation, ie $S_i = \alpha S_{i-1} + \mathrm{ADC}_i$
- Shown above is $\alpha = 0.99$. Horizontal track is rescued, pedestal under control
- Disadvantage for online implementation is need for floating-point (or fixed-point approximation)